A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks

Lan Wang and Yang Xiao
Department of Computer Science
The University of Memphis
373 Dunn Hall
Memphis, TN 38152 USA
Emails: lanwang@memphis.edu, yangxiao@ieee.org

***Abstract*— Sensor networks have a wide range of potential, practical and useful applications. However, there are issues that need to be addressed for efficient operation of sensor network systems in real applications. Energy saving is one critical issue for sensor networks since most sensors are equipped with non-rechargeable batteries that have limited lifetime. To extend the lifetime of a sensor network, one common approach is to dynamically schedule sensors' work/sleep cycles (or duty cycles). Moreover, in cluster-based networks, cluster heads are usually selected in a way that minimizes the total energy consumption and they may rotate among the sensors to balance energy consumption. In general, these energy-efficient scheduling mechanisms (also called topology configuration mechanisms) need to satisfy certain application requirements while saving energy.**

In this paper, we provide a survey on energy-efficient scheduling mechanisms in sensor networks that have different design requirements than those in traditional wireless networks. We classify these mechanisms based on their *design assumptions* and *design objectives*. Different mechanisms may make different assumptions about their sensors including detection model, sensing area, transmission range, failure model, time synchronization, and the ability to obtain location and distance information. They may also have different assumptions about network structure and sensor deployment strategy. Furthermore, while all the mechanisms have a common design objective to maximize network lifetime, they may also have different objectives determined by their target applications.

Keywords: Sensor Network, Energy Consumption, Sleep-mode, Scheduling, Energy-Efficient Scheduling

I. Introduction

Sensor networks have a wide variety of applications in both military and civil environment [1]. Some of these applications, e.g., natural habitat monitoring, require a large number of tiny sensors and these sensors usually operate on limited battery power. According to [9], individual sensors can last only 100-120 hours on a pair of AAA batteries in the active mode. On the other hand, since the number of sensors is huge and they may be deployed in remote, unattended, and hostile environments, it is usually difficult, if not impossible, to recharge or replace their batteries. This problem is compounded by the fact that battery capacity only doubles in 35 years [24].

Since a sensor network is usually expected to last several months to one year without recharging [22], [30], optimal energy consumption, i.e., minimizing energy consumed by sensing and communication to extend the network lifetime, is an important design objective. In the meantime, how well a sensor network can collect sensory data depends on its sensing coverage and network connectivity. Therefore, maintaining sufficient sensing coverage and network connectivity are important design requirements for sensor networks. Furthermore, fault tolerance should also be considered when minimizing energy consumption in the presence of individual sensor failure [17]. In fact, there are many design objectives for a sensor network, some of which are summarized in Section II-B.

To minimize energy consumption and extend network lifetime, a common technique is to put some sensors in the sleep mode and put the others in the active mode for the sensing and communication tasks. When a sensor is in the sleep mode, it is shut down except that a low-power timer is on to wake up the sensor at a later time [12], therefore it consumes only a tiny fraction of the energy consumed in the active mode ([9], [22]). Moreover, in cluster-based networks, cluster heads are usually selected in a way that minimizes the total energy consumption and they may rotate among the sensors to balance energy consumption.

Both approaches try to save energy by configuring the sensors into certain topologies, therefore such mechanisms have been referred to as "topology configuration" mechanisms. In this paper, we call them "energy-efficient scheduling mechanisms". There are many other methods to save energy, such as reducing communication cost and reducing control messages, but we only discuss distributed scheduling mechanisms in this paper. For example, several MAC layer power saving schemes (e.g., [23] and [33]) reduce energy consumption by minimizing radio transceivers' idle time. More specifically, a radio will be turned off when it is not actively sending and receiving, and it will be turned on when communication is expected. These schemes are different from and complementary to the surveyed mechanisms because they can be used to fine tune the communication cost after a topology is configured.

Although there are many scheduling mechanisms published in the literature, different mechanisms often have different assumptions, mainly because they are considered in the context of different applications. The design assumptions include, but are not limited to, detection model, sensing area, transmission range, failure model, time synchronization, location information, and distance information. There are also different assumptions about network structure and sensor deployment strategy. Furthermore, while all the mechanisms have a common design objective to maximize network lifetime, they may also have different objectives determined by their target applications. For example, a surveillance application may require the working sensors to achieve a certain degree of sensing coverage. Other design objectives include network connectivity, high data delivery ratio, high quality of surveillance, stealthiness, balanced energy consumption, scalability, robustness, and simplicity.

It is unfair and sometimes misleading to compare the scheduling mechanisms without considering the different assumptions and objectives, yet they are often compared without taking into consideration these factors. Therefore, we believe that the first step to understand the differences among the mechanisms is to understand their assumptions and objectives. To this end, we have surveyed 15 distributed scheduling mechanisms designed for sensor networks and present the results in this paper. This is by no means a complete survey of the literature. Rather, we hope that this set of sample mechanisms could help us understand the wide range of design choices.

The rest of the paper is organized as follows. Section 2 discusses the design assumptions and objectives. Section 3 explains the different energy consumption modes that a sensor can use. Section 4 and Section 5 review the scheduling mechanisms in non-hierarchical networks and hierarchical networks, respectively. After understanding all the schemes, we provide a classification in Section 6. We conclude this survey in Section 7. Finally, some theoretical results are summarized in Appendix A.

II. Classification Methodology

All the surveyed mechanisms have a common objective – maximizing sensor network lifetime. However, they make quite different assumptions regarding the sensors and the sensor network. They also have different objectives that are determined by their applications. Therefore, a fair comparison among the surveyed mechanisms has to take into consideration these factors. In this section, we summarize the design assumptions and objectives. Based on this information, any classification can be easily derived. We will present the result of our classification in Section 6 after reviewing all the mechanisms.

A. Design Assumptions

Since the focus of this survey is energy saving, all the surveyed mechanisms make the following *common* assumptions: (1) each sensor has limited energy supply and (2) the sensor network is expected to run for a long time. Below we discuss the *different* design assumptions that reflect different network structures, sensor deployment strategies, and sensor capabilities (see Figure 1).

a) Network Structure: A sensor network can be *non-hierarchical* or *flat* in the sense that every sensor has the same role and functionality. Alternatively, a sensor network can be *hierarchical*. For example, in sensor networks designed for detection and tracking, some sensors may be designated as the fusion centers: they collect the reports from the sensors in their neighborhood, make a decision regarding whether an object has been detected, and send a report to the base station. These networks are often *cluster-based* (or sentry-based) in which the cluster heads (or sentries) have a more prominent role than the other sensors. Several surveyed mechanisms, e.g., [12], assume that *the sensor network is cluster-based*. The detailed survey in Section 4 and Section 5 is in fact organized (at the top level) based on whether the mechanism makes this assumption or not.

b) Sensor Deployment Strategy: The performance of a sensor network, e.g., its sensing coverage, can be affected by how the sensors are initially placed. There are various *sensor placement mechanisms*. For example, sensors may be dispersed from an airplane flying over the sensing field, and they may also be manually placed at selected locations. In the first scenario, the sensors' locations are likely to follow a random distribution. Several surveyed mechanisms [30], [32] assume that *the sensors are randomly and uniformly distributed over the sensing field*. Sometimes a mechanism does not explicitly state this assumption, but its performance is best when this assumption holds. Some papers also use the two-dimensional Poisson distribution [22]. In addition, a few mechanisms assume that the sensors form a grid [22], [27].

Most of the surveyed mechanisms assume that there are some redundant sensors in the network that can be turned off. One explicitly assumes that the total number of sensors is orders of magnitude higher than the number of working sensors [32]. We consider this level of density to be "high". Otherwise, if the two numbers are on the same order, we consider the level of density to be "normal".

c) Detection Model: Most surveyed mechanisms assume that a sensor can detect an object as long as the object is inside its sensing range, i.e., *the detection model is deterministic*. However, one notable exception is [31] – it uses a probabilistic detection model in which the detection probability of an object is a function of the distance between the object and the sensor.

d) Sensing Area: The sensing area is usually assumed to be either a circular area or a 3-D sphere. Moreover, the sensors are usually assumed to have the same sensing range. There are several mechanisms that are extensible to any convex and non-uniform (but still deterministic) sensing areas [4].

e) Transmission Range: Several mechanisms ([12], [32]) assume *that a sensor's radio transceiver is capable of changing its transmission power in continuous steps to achieve different transmission ranges*. Some sensors, such as the MICA2 mote [8], provide multiple levels of transmission power. However, the actual transmission range under a particular power level may also be affected by many external factors such as the height of the sensor and its surrounding objects [15].

f) Time Synchronization: Several mechanisms assume that *sensors are time synchronized* so that they can wake up at the same time to start a new round of scheduling [22], [28], [34]. Many time synchronization algorithms have been proposed for sensor networks (see [13] for an example).

g) Failure Model: How nodes may fail is an important assumption about both the sensors and the environment in

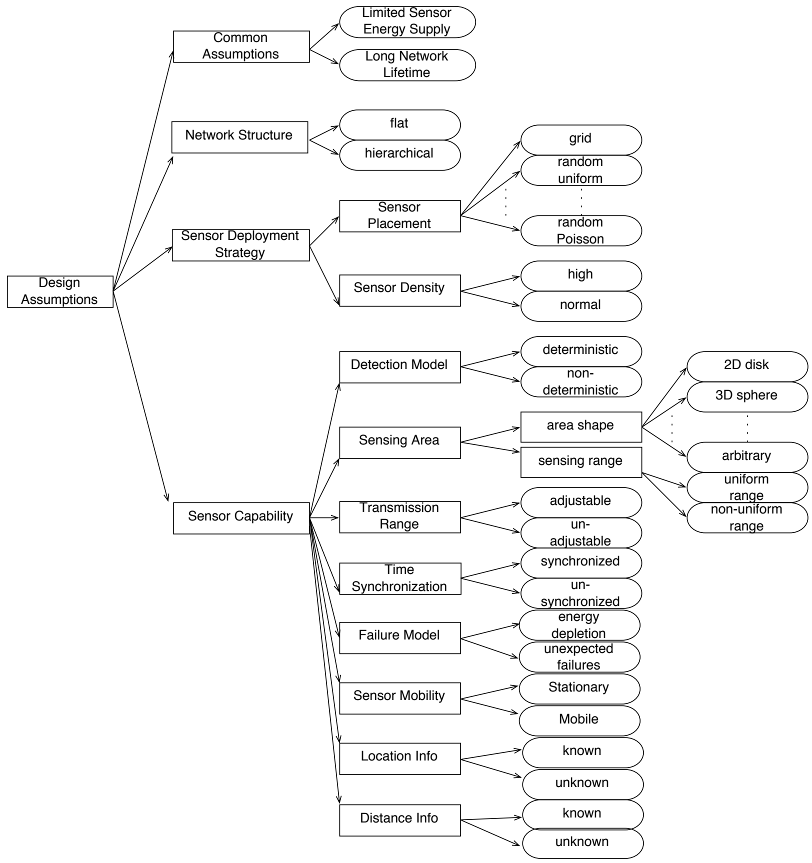

Fig. 1. Design Assumptions of Scheduling Mechanisms

which they are deployed. All the surveyed mechanisms assume that a sensor fails when its energy is depleted. Several mechanisms, e.g., [32], further assume that sensors may fail unexpectedly *before* they run out of battery. For example, the sensors may be destroyed by tanks if they are spread in a military field.

h) Sensor Mobility: All the surveyed mechanisms either assume explicitly that *the sensor network is stationary* or do not make an explicit assumption about mobility. In fact, several papers, e.g., [30], argue that most real-world sensor networks involve little or no mobility.

i) Location Information: Several surveyed mechanisms assume that *sensors can identify their geographic locations*. The location information is usually used to determine whether (and how much) a node's sensing area overlaps with its neighbors' sensing areas. If the location for each sensor is predetermined and the sensors are not mobile, then the location information can be hard-coded in the sensors before they are deployed. Otherwise, sensors may need to be equipped with GPS devices or run a localization algorithm such as the one proposed in [5].

j) Distance Information: [10] and [12] assume that *sensors in a cluster-based sensor network can determine the distance to their cluster head*. Distance information can be obtained via location information (but the reverse is not true). In addition, distance information may be inferred from the strength of received signals [12].

B. Design Objectives

Applications differ in their requirements, therefore the underlying sensor networks usually have different design objectives or have different priorities among the objectives. *Maximizing network lifetime* is certainly one of the most important design objectives for all the sensor networks that need to run for a long time. A sensor network, however, is built to accomplish certain tasks, e.g., to perform sensing and deliver sensory data. Therefore, one or more *Quality of Service objectives*, such as maintaining sensing coverage, are usually considered along with minimizing energy consumption. Furthermore, a design may consider *high-level objectives* such as robustness, scalability and simplicity. Since a design decision for achieving one objective may have an impact on some of the other objectives, we not only summarize the objectives in this section, but also discuss the relationship among them.

a) Maximizing Network Lifetime: Network lifetime has been defined in various ways [6], [12], [30], [32], [15] and an energy-efficient mechanism may choose to maximize a certain type of network lifetime. In the simplest case, a network may be considered alive when any of the sensors is alive. Network lifetime can also be calculated as the duration of time when the percentage of sensors that have depleted their energy is below a threshold, e.g., 90% [6], [12]. Alternatively, one or more quality of service measures can be taken into account. For example, a network may be considered functional only when its sensing coverage (or degree of connectivity, data delivery

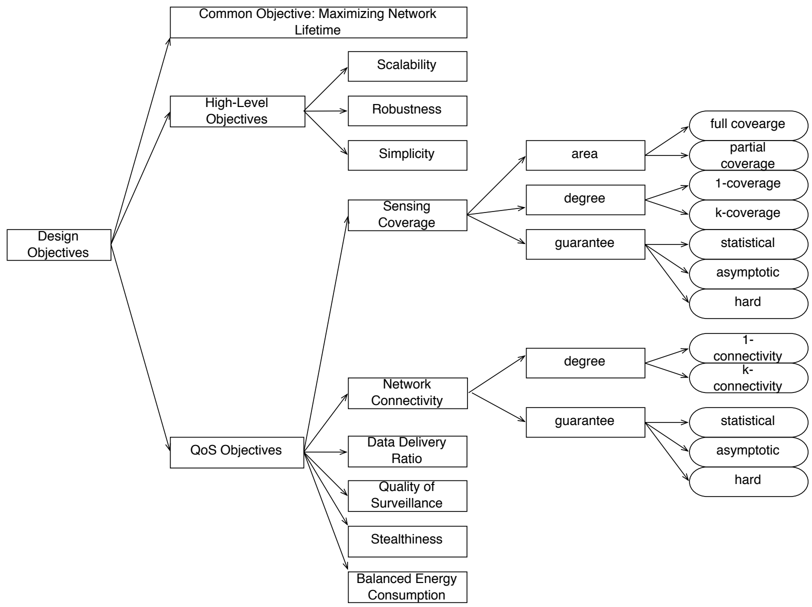

Fig. 2. Design Objectives of Scheduling Mechanisms

ratio, etc.) is above a certain threshold [30], [32].

b) Sensing Coverage: Since sensing is the primary function of a sensor network, sensing coverage is an important QoS measure of the network. If every point in a field is monitored by at least 1 sensor, the sensor network is said to achieve 1-*coverage*. If every point in the field is monitored by at least K sensors, the sensor network is said to achieve K-*coverage* [32] (1-coverage is a special case of K-coverage). A sensor network may also provide *partial* 1-coverage or *partial* K-coverage. Sometimes a deterministic guarantee may not be possible under the chosen assumptions. Therefore, a few mechanisms ensure *asymptotic coverage* when the number of sensors goes to infinity [22].

c) Network Connectivity: If sensory data need to travel multiple hops to reach the destination (e.g., the base station), it is important to maintain the connectivity among the sensors. Some mechanisms, e.g., [29], can even configure the network to a specific *degree of connectivity* required by the application, given a sufficiently high sensor density. Similar to sensing coverage, connectivity can also be achieved in an *asymptotic* sense, i.e., the network is guaranteed to be connected when there is an infinite number of sensors [32].

d) Data Delivery Ratio: A high data delivery ratio is another desirable QoS objective for some applications. It can be measured by the average percentage of data delivered from a source to a sink. However, this measure is not appropriate when there is in-network data aggregation.

e) Quality of Surveillance: Gui and Mohapatra proposed a metric, *Quality of Surveillance (QoSv)*, to measure the performance of target-tracking sensor networks [14]. QoSv is defined as the *inverse of the average distance traveled by a target before it is detected by a sensor*. This definition implies that if a sensor network can detect a moving target within a shorter distance following the target's intrusion than other networks, it is considered to have a higher quality of surveillance. QoSv is not purely a function of sensing coverage. It also depends on other factors such as the sensors' geographical distribution. If the sensors are clustered in certain areas, then targets entering the network from the other areas can travel a long distance before they are detected. Therefore, the average distance traveled by a random target before its detection in this network may be longer than in another network that has the same sensing coverage but whose sensors are more evenly distributed over its entire area.

f) Stealthiness: Certain surveillance applications require the operations of the sensor network to be stealthy or less likely to be detected by others. Stealthiness can be achieved by reducing the number of control messages. Shortening the communication period may also improve stealthiness, if the sensors communicate with each other only during that period [15].

g) Balanced Energy Consumption: Some mechanisms strive to balance the energy consumption among the sensors. One common argument for doing this is that if the energy of certain nodes is depleted before the others, holes may appear in the sensing coverage or the sensor network may become disconnected prematurely. A counter argument is based on the assumption of high sensor density (see Section II-A for the design assumptions): even if those nodes die prematurely, there will still be some redundant nodes that can be turned on at or near those locations.

h) Scalability: There is no universal definition of scalability. However, it is generally undesirable for sensors to have a state overhead or computation overhead that increases linearly or even faster with the number of all potential neighbors. On the other hand, if each node needs to keep track of only *active* neighbors and the number of active neighbors is small at any given time, the mechanism may still be considered scalable.

i) Robustness: Robustness is the ability of a network to withstand unexpected failures. For example, sensor nodes on a battle field may be destroyed by tanks and bombs before their batteries run out. A robust mechanism cannot expect everything to go as planned. For an instance, it cannot expect all the sleeping nodes to wake up – some of these nodes may have stopped functioning while they were sleeping. Obviously, the assumptions made by the designers regarding possible failures can have a significant impact on the robustness of a mechanism.

j) Simplicity: Current sensors have very limited memory space for storing programs, e.g., the MICA2 mote has only 8KB of memory for this purpose. Moreover, they usually have limited computation power and they are difficult to debug. Therefore, simpler mechanisms are more likely to be deployed in sensor networks.

k) Relationship between Design Objectives: Since it is impossible to enumerate all the possible relationships among the design objectives, we list some of the important ones here: (1) when sensors' transmission range is at least twice their sensing range, ensuring K-coverage can lead to K-connectivity [29], [34]. However, the reverse is not necessarily true; (2) a higher degree of connectivity generally leads to a higher degree of robustness against failures, since more links need to be removed to make the network disconnected; (3) data delivery ratio normally improves as the degree of connectivity increases. However, if the connectivity degree is too high, the data collisions among sensors may adversely affect the delivery ratio; (4) a higher degree of stealthiness may indicate lower energy consumption and improved data delivery ratio, if stealthiness is achieved through minimizing control messages; (5) a simpler mechanism tends to be more robust because the programmer is less likely to make a mistake.

III. ENERGY SAVING MODES OF SENSORS

In order to understand the surveyed mechanisms, one first needs to be able to differentiate the various energy saving modes that can be provided by a sensor. One complexity here is that different types of sensors may support different sets of modes and even if they support the same set of modes, they often use different terminology. To make our presentation clear, we define the major modes of a sensor as follows:

- **On-Duty**: all the components in the sensor are turned on. The sensor is able to collect sensory data, send/receive messages, process data and messages, and do other types of computation. This mode is also called the **active** mode in the literature. It is *not* an energy-saving mode.
- **Sensing Unit On-Duty**: at least one sensing unit and the processor are turned on, but the transceiver is turned off. In this mode, the sensor is capable of sensing and processing sensory data, but not transmitting or receiving messages. We also use the shorter form **SU-On-Duty** in the paper.
- **Transceiver On-Duty**: the transceiver and the processor are turned on, but all the sensing units are turned off. In this mode, the sensor is capable of transmitting, receiving and processing messages, but not sensing. We also use the shorter form **TR-On-Duty** in the paper.
- **Off-Duty**: the sensor's processor is turned off, but a timer or some other triggering mechanism may be running to wake up the sensor. This mode is also called the **Sleep** mode in the literature.

Note that some sensors have multiple Off-Duty (Sleep) modes, each with a different wakeup mechanism. For example, the μAMPS sensor has three sleep modes: **Monitor**, **Observe**, and **Deep Sleep** [26]. The processor is turned off in all three modes, so the sensor cannot process any sensory data or messages. However, in the *Monitor* mode, both the sensing unit and the transceiver are left on to receive wakeup signals. In the *Observe* mode, only the sensing unit is on. Note that the *Observe* mode is different from the *SU-On-Duty* mode as the processor is turned on in the latter. In the *Deep Sleep* mode, neither the sensing unit nor the transceiver is turned on, so the sensor relies on a preset internal timer to wake itself up. Most sensors provide a sleep mode similar to μAMPS' Deep Sleep mode. Note that it is possible to design a fourth sleep mode that we call the **Listen** mode. In this mode, only the transceiver is turned on to receive wakeup signals (but the processor and sensing unit are off).

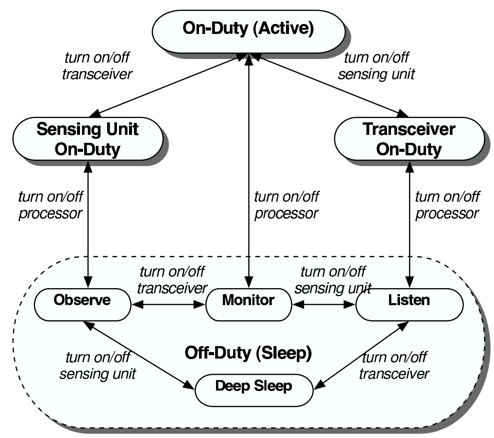

Fig. 3. Transitions between Different Sensor Modes

In figure 3, we show how a sensor can change from one mode to another. A distributed scheduling mechanism allows each sensor to determine when it should switch its mode and what mode it will switch to. Such a mechanism should minimize the overall energy consumption and achieve the application-specific objectives. In the rest of this section, we compare the energy consumption of different sensor modes.

Sensors consume the most energy in the *On-Duty* mode and save the most energy in the *Deep Sleep* mode. The energy consumption in the other modes depends on the design of the sensors, but in general one can expect the following: the energy consumption of the *Observe* and *Listen* modes should be less than that of the *Monitor* mode which in turn should be less than that of the *SU-On-Duty* and *TU-On-Duty* modes. Moreover, sensor generally save more energy in the *SU-On-Duty* mode than in the *TR-On-Duty* mode, because communication usually consumes more energy than sensing ([8], [18]). However, the energy consumption in the TR-On-Duty mode also depends on the actual communication pattern and frequency. First, the energy cost of transmitting messages may be quite different from that of receiving messages. Normally, transmitting messages costs more energy than receiving messages, but there are exceptions ([8], [26]). For example, the MICAz sensor draws 19.7mA of current when receiving a message and 17mA of current when transmitting at the 1mW level (the highest transmission power) [8]. Second, more frequent communication may consume more energy. Although idling may consume a similar level of energy as receiving (or even sending) a message ([30], [32]), this may not apply to all sensors. Moreover, frequent communication can cause collisions and retransmissions, which will lead to more energy consumption.

Almost all of the surveyed mechanisms take advantage of the energy saving feature of the *Deep Sleep* mode. However, the designers of μAMPS proposed a mechanism to take advantage of all of its three sleeping modes [26]. Some mechanisms also exploit the *SU-On-Duty* and *TR-On-Duty* modes. Note that sometimes it is not easy to identify which energy-saving mode that a mechanism uses. For example, in [4], sensors cannot be completely turned off, but they can enter an *idle* mode in which they are only required to listen to messages. However it is not clear whether this idle mode refers to the *Monitor* mode of μAMPS (an off-duty mode) or the *TR-On-Duty* mode.

IV. Distributed Scheduling Mechanisms in Non-hierarchical Networks

In this section, we review nine scheduling mechanisms that can be applied to non-hierarchical sensor networks. They may also be used for hierarchical networks, e.g., within each cluster.

A. Random Independent Scheduling (RIS)

In [22], Kumar et al. adopt the Randomized Independent Scheduling (RIS) mechanism to extend network lifetime while achieving asymptotic K-coverage. RIS assumes that time is divided into cycles based on a *time synchronization* method. At the beginning of a cycle, each sensor independently decides whether to become active with probability p or go to sleep with probability $1 - p$. Thus the network lifetime is increased by a factor close to $\frac{1}{p}$ (i.e., p determines the network lifetime).

Furthermore, Kumar et al. derived the conditions for *asymptotic K-coverage* when RIS is used with three different sensor deployment strategies – grid, random uniform, and 2-dimensional Poisson (see Appendix C). Their results can be applied in several ways. First, the number of sensors that should be initially deployed can be determined in order to ensure asymptotic K-coverage. Second, the number of additional sensors needed or the new value of p when dynamically reconfiguring the network to a different degree of coverage can be calculated. Note that these results only apply to the RIS mechanism.

RIS is a very simple self-scheduling mechanism. It has the following major characteristics: (1) it does not require location or distance information; (2) it has no communication overhead; (3) nodes do not maintain a neighborhood table; (4) because the sensors do not dynamically evaluate their situation, the basic RIS mechanism is not robust against unexpected failures that destroy the sensors before they run out of energy. One (obvious) solution is to let the base station periodically evaluate the network performance; whenever the required QoS is not met, the base station can reconfigure the sensors' parameters or increase the sensor density (e.g., based on Kumar et al.'s results).

B. Sponsored Sector

Tian and Georgana proposed a distributed scheduling mechanism to save energy while preserving sensing coverage [28]. To avoid reducing sensing coverage, this mechanism allows a sensor to turn off only if its sensing area is *completely* covered by its neighbors' sensing areas – the neighbors are called this node's *off-duty sponsors* in this case and the sector that a neighbor covers in its sensing area is called a *sponsored sector*. Each sensor uses its neighbors' location information and sensing range to determine the sponsored sectors and concatenate the central angles of the sponsored sectors. If the entire 360 degrees of the central angle are covered, then the node is eligible to enter the off-duty mode. We refer to this mechanism as *Sponsored Sector* in the rest of this paper.

Note that this mechanism only considers those neighbors within a node's sensing area to be potential off-duty sponsors, i.e., neighbors in the $(r,\, 2r)$ range are ignored even if their coverage may overlap with this node's sensing area, so Sponsored Sector may underestimate the number of sensors that can be turned off.

The operation of Sponsored Sector is divided into rounds. In each round, the sensors start in a *self-scheduling phase* in which they obtain neighbors' location information and decide whether to turn off. Then the on-duty sensors enter a sensing phase in which they gather sensing data. Obviously, the energy saving depends on how long the self-scheduling phase is compared to the duration of each round; the quicker the sensors stabilize in the self-scheduling phase, the more energy will be saved.

The Sponsored Sector mechanism also uses a random *back-off* strategy to prevent multiple sensors from taking actions simultaneously. More specifically, before a sensor turns off, it enters a *ready-to-off* state and sets a random timer. The ready-to-off state in this scheme can be either the on-duty mode or the TR-on-duty mode defined in Section III. In this state, if the sensor receives a message from another sensor indicating that the latter is about to turn off, it re-evaluates its off-duty eligibility. If it becomes ineligible, it will enter the *on-duty* state. When the timer expires, the sensor will enter the *off-duty* state. Note that the on-duty sensors will stay active until the end of a round.

In summary, the Sponsored Sector mechanism demonstrates that it is possible to preserve sensing coverage with geographic location information. It has the following major characteristics: (1) nodes need accurate location information; (2) nodes are time-synchronized so that they know the beginning of each round; (3) there is a message overhead for advertising location information and scheduling (but only at the beginning of each round); (4) nodes maintain per-neighbor state to keep track of the number of active neighbors; (5) in each round, working nodes never go back to sleep. However, the set of working nodes may be different in different rounds, so energy consumption may still be balanced among the nodes; (6) the off-duty eligibility rule is relatively conservative compared to some of the other mechanisms, so its energy saving may not be as high as those mechanisms.

C. Maximization of Sensor Network Life (MSNL)

Berman et al. [4] formulated the sleep-scheduling problem as a maximization problem with constraints on battery lifetime and sensing coverage. They also presented a centralized and a distributed algorithm to maximize network lifetime while achieving K-coverage. Their distributed mechanism can guarantee a *specific degree* of sensing coverage (assuming that the sensor density is high enough) whereas the aforementioned Sponsored Sector mechanism preserves the *existing* coverage degree. In this mechanism, each sensor is in one of three states: *active*, *idle* or *vulnerable*. In the vulnerable state, if the sensor discovers that part of its sensing area cannot be covered by any of its active or vulnerable neighbors, it immediately enters the *active* state. Otherwise, it enters the *idle* state if its sensing area can be monitored by either active neighbors or vulnerable neighbors with a higher energy level.

In summary, the above mechanism can guarantee a *specific degree* of sensing coverage. Below are the major characteristics of this algorithm: (1) nodes need accurate location information; (2) although nodes are not time-synchronized, they have semi-synchronous monitoring schedules (due to global reshuffles); (3) nodes need to broadcast their state and energy level in addition to their location; (4) nodes cannot completely turn themselves off in the idle state; (5) unlike the Sponsored Sector mechanism, there is no random backoff in this mechanism. Therefore, it is possible for multiple neighboring sensors to enter the idle state simultaneously.

D. Lightweight Deployment-Aware Scheduling (LDAS)

In [30], Wu et al. proposed a distributed scheduling mechanism called LDAS (Lightweight Deployment-Aware Scheduling). Unlike previous studies, this work assumes that sensor nodes are *not* equipped with GPS or other devices to obtain location information. Since it is difficult, if not impossible, to determine whether a node's sensing area is *absolutely* covered by other nodes without location information, the goal here is to provide *statistical* guarantees on sensing coverage. Note that [32] does not assume the knowledge of location information either, but this work is complementary to [32] as the latter guarantees *asymptotic network connectivity*. LDAS assumes that each working node has a mechanism to know the number of working nodes in its neighborhood. When the number of working neighbors exceeds a threshold determined by the application's requirement on sensing coverage (see Appendix for how the threshold is calculated), the node randomly selects some of its neighbors to turn off and sends tickets to them. When a node collects enough tickets from its neighbors, it may enter the off-duty mode after a random back-off period.

In summary, LDAS can achieve a specific level of partial sensing coverage in a *statistical* sense. Below are the major characteristics of LDAS: (1) nodes are assumed to be randomly and uniformly distributed over the coverage area; (2) unlike the Sponsored Sector and MSNL mechanisms, LDAS does not require accurate location information; (3) each node needs to know how many sensors are within its sensing range; (4) unlike the previous three mechanisms, nodes have asynchronous sleeping schedules; (5) energy consumption is balanced among the nodes since the longer a node works, the more tickets it may accumulate and the more likely it will be turned off.

E. Probing Environment and Adaptive Sensing (PEAS)

Ye et al. developed a mechanism called PEAS (Probing Environment and Adaptive Sensing) that can extend the lifetime of a high-density sensor network in a harsh environment [32]. What distinguishes this work from the previous studies are first its assumptions. First, it assumes that sensor nodes may fail frequently and unexpectedly, which makes synchronized sleeping algorithms infeasible because they depend on the predictability of sensors' lifetime. Second, it assumes that the sensor network is so dense that the total number of sensors may be orders of magnitude higher than the number of working nodes. As a result, it is infeasible for nodes to maintain per-neighbor state. Finally, it assumes that nodes do not have location information. The authors argue that these assumptions lead to a design that is more robust against failures and easier to implement in a real sensor network. PEAS conserves energy by separating all the working nodes by a minimum distance of c. To check if there is a working neighbor nearby, each node broadcasts a message (probe) with a transmission range of c after sleeping for a random period. A node will enter the on-duty mode only if it receives no replies from working neighbors; otherwise it will stay in the off-duty mode. In the same paper, Ye et. al. proved that PEAS can guarantee *asymptotic connectivity* as long as the sensor network satisfies two conditions on the sensor density and the probing range.

In summary, PEAS achieves asymptotic network connectivity. Below are the major characteristics of PEAS: (1) nodes are assumed to be randomly and uniformly distributed; (2) similar to LDAS, nodes do not need accurate location information; (3) similar to LDAS, nodes have asynchronous schedules; (4) unlike most of the surveyed mechanisms, nodes do not maintain per-neighbor state; (5) working nodes never go back to sleep, which may result in unbalanced energy consumption; (6) nodes adapt their probing rate to control the overall message overhead.

F. Optimal Geographic Density Control (OGDC)

Zhang and Hou proved that 1-coverage implies 1-connectivity when the ratio between the radio transmission range and the sensing range is at least two [34] (also see Appendix E). Assuming that this condition is satisfied, Zhang and Hou further proposed a distributed mechanism, Optimal Geographic Density Control (OGDC), to maximize the number of sleeping sensors while ensuring that the working sensors provide complete 1-coverage and 1-connectivity[34]. OGDC tries to minimize the overlapping area between the working sensors. A sensor is turned on only if it minimizes the overlapping area with the existing working sensors and if it covers an intersection point of two working sensors. A sensor can verify whether it satisfies these conditions using its own location and the working sensors' locations. OGDC's protocol is quite similar to that of the Sponsored Sector mechanism, except that they use different on-duty/off-duty eligibility rules and the Sponsored Sector mechanism is more conservative when turning off sensors.

In summary, OGDC can maintain both 1-coverage and 1-connectivity when the radio transmission range is at least twice the sensing range. It has the following major characteristics: (1) nodes need accurate location information; (2) nodes need to maintain time synchronization; (3) there is message overhead for advertising location information and scheduling *only at the beginning of each round*; (4) in each round, working nodes never go back to sleep, but different nodes may be working in different rounds so energy consumption may still be balanced among all the nodes.

G. Coverage Configuration Protocol (CCP)

Wang et al. proposed an integrated coverage and connectivity configuration protocol called CCP [29]. This protocol aims to maximize the number of sleeping nodes, while maintaining both K-coverage and K-connectivity. Note that the OGDC mechanism ensures 1-coverage and 1-connectivity. CCP's capability is based on the theorem that K-coverage implies K-connectivity when the transmission range is at two times the sensing range (see [29] and Appendix F of this paper). To ensure K-coverage, a node only needs to check whether the intersection points inside its sensing area are K-covered (based on a theorem proved in [29]).

Since CCP cannot guarantee network connectivity when the radio transmission range is less than twice the sensing range, the authors combined CCP with SPAN [7], a connectivity preserving scheduling mechanism, to achieve their dual objectives in this scenario. Specifically, a node will sleep only if it satisfies the eligibility rules in both CCP and SPAN. In other words, even if the node is redundant with regard to its sensing coverage, it may still stay active to maintain network connectivity. Note that SPAN cannot configure a network to a specific degree of connectivity, but it will preserve the original connectivity of the network after turning off some redundant nodes.

Nodes running CCP are in one of three modes: ACTIVE, LISTEN and SLEEP. Each node is initially in the ACTIVE mode and when it receives a message, it determines whether it should go to SLEEP. If so, it enters the LISTEN mode and starts a random timer (the LISTEN mode could be either the on-duty mode or the TR-on-duty mode in our terminology). When this timer expires and if the node is still eligible to sleep, it will enter into the SLEEP mode. Otherwise, it will stay in the LISTEN mode. In the SLEEP mode, a node will also set a random timer. When the timer expires, it will enter the LISTEN mode and check if it is still eligible to sleep. If so, it will go back to sleep. Otherwise, it will enter the ACTIVE mode.

In summary, CCP can configure a network to a specific degree of sensing coverage and a specific degree of connectivity required by the application when $R_c \geq 2R_s$; CCP+SPAN can configure a network to a specific sensing coverage and preserve its original connectivity when $R_c < 2R_s$. Below are the major characteristics: (1) CCP requires accurate location information; (2) each node needs to maintain a neighborhood table; (3) nodes have asynchronous sleep schedules; (4) working nodes may go back to sleep, so that the energy consumption is balanced among all the nodes.

H. Adaptive Self-Configuring sEnsor Networks Topologies (ASCENT)

In [6], Cerpa and Estrin proposed using sensors' local measurements to automatically configure network topology in a high density sensor network. The goal is to maintain a certain data delivery ratio while allowing redundant sensors to stay asleep in order to conserve energy. Achieving this goal requires configuring the network to the right level of connectivity; it cannot be too low to hamper data delivery, but it cannot be too high either since neighboring nodes might interfere with each other leading to a high collision rate. The approach adopted by ASCENT is to let sensors measure their connectivity as well as their data loss rate and activate their neighbors based on these local measurements.

ASCENT is similar to PEAS in several ways. First, ASCENT assumes that there is a high density of sensor nodes. Second, after the sensors are activated, they never go back to sleep – fairness in energy consumption is not an important design issue here. Third, both are decentralized mechanisms that allow sensors to use locally measured information to adjust network topology. However, unlike PEAS, ASCENT does not guarantee network connectivity in any sense (the network could be partitioned), although the delivery of data indicates that there is a certain level of connectivity.

In summary, ASCENT allows sensors to automatically adapt their connectivity to achieve a required level of data delivery ratio. It has the following major characteristics: (1) nodes do not need accurate location information; (2) there is no periodic message overhead for neighbor discovery; (3) nodes maintain per-neighbor state to keep track of the number of active neighbors; (4) working nodes never go back to sleep, which may result in unfair energy consumption.

I. Probing Environment and Collaborating Adaptive Sleeping (PECAS)

PECAS is proposed by Gui and Mohapatra in [14]. It is an extension of the PEAS protocol proposed in [32]. It has the same environment probing mechanism as PEAS, but it does not let working nodes stay awake indefinitely. The designers argue that "the failure of nodes (due to energy depletion) may cause partitioning of the network or isolation of nodes ... it is desirable to balance energy consumption among the neighboring nodes." Therefore, a working node in PECAS will go back to sleep after a specified period of time. It also advertises the remaining working time in its reply messages to its neighbors' probe messages. In this way, a working neighbor who decides to enter the off-duty mode can schedule itself to wake up before this node goes to sleep, thus preventing the occurrence of blind spots.

Simulation results in [14] show that PECAS can achieve a higher QoSv than PEAS. However, the energy saving of PECAS is lower than that of PEAS due to the higher message exchange overhead. Since PECAS is similar to PEAS in all the other aspects, please refer to Section IV-E for a summary of the major characteristics of PEAS.

V. Distributed Scheduling Mechanisms in Hierarchical Networks

In a hierarchical network such as a cluster-based network, sensors organize themselves into clusters and each cluster has a cluster head. The cluster heads may or may not be more powerful than the other sensors. Each cluster head manages the sensors in its own cluster for communication between the cluster and the base station. Communication between cluster heads and the base station may be multi-hop through other cluster heads. In this section, we introduce six cluster-based scheduling mechanisms as follows.

A. Low-Energy Adaptive Clustering Hierarchy (LEACH)

Heinzelman et al. [17] proposed Low-Energy Adaptive Clustering Hierarchy (LEACH), a cluster-based protocol utilizing randomized rotation of cluster heads to evenly distribute work load among the sensors. In LEACH, the operation is divided into cycles and each cycle includes a *set-up* phase and a *steady* phase. During the set-up phase, cluster heads are selected and each sensor joins a cluster by choosing the cluster head that requires the minimum communication energy. During the steady phase, each cluster head aggregates the data from the sensors in its cluster and then transmits the compressed data to the base station.

Since cluster heads have more responsibilities and consume more energy, LEACH let different sensors become cluster heads in each cycle to prevent the cluster heads from running out of energy first. Cluster heads are self-elected at the beginning of each cycle. In the r-th cycle, a sensor that has not become a cluster head during the previous $1/P$ cycles decides to become a cluster head with the probability $P\big/\big(1 - P \times \big(r \bmod \frac{1}{P}\big)\big)$. This probability will ensure that on average a percentage P of the sensors become cluster heads and that existing cluster heads have a lower likelihood to become cluster heads in the next round. *To conserve energy, non-head sensors are turned off at all times except during their transmission time*.

In summary, LEACH reduces energy consumption by (a) minimizing the communication cost between sensors and their cluster heads and (b) turning off non-head nodes as much as possible. It has the following major characteristics: (1) it rotates the cluster heads in a randomized fashion to achieve balanced energy consumption; (2) sensors have synchronized clocks so that they know the beginning of a new cycle; (3) sensors do not need to know location or distance information; (4) the time duration of the set-up phase is non-deterministic, and if the duration is too long due to collisions, sensing services are interrupted; in such cases, LEACH may be unstable during the set-up phase depending on the density of sensors;

B. Enhanced Low-Energy Adaptive Clustering Hierarchy (E-LEACH)

E-LEACH [16] further improved LEACH in two major aspects. First, the authors proposed a cluster head selection algorithm for sensor networks that have non-uniform starting energy level among the sensors. However, this algorithm assumes that sensors have global information about other sensors' remaining energy. Second, the authors determined that, under certain assumptions, the required number of cluster heads has to scale as the square root of the total number of sensor nodes to minimize the total energy consumption. Since the other aspects of E-LEACH are the same as LEACH, we do not summarize them here.

C. Minimizing Communication Cost in Hierarchically-Clustered Networks

Bandyopadhyay et al. [2], [3] considered a simple strategy to select cluster heads – they are chosen randomly with a probability p. There are two kinds of cluster heads: volunteer cluster heads and forced cluster heads. Each sensor can become a volunteer cluster head with probability p. A volunteer cluster head advertises itself to the neighboring sensors, which then forward the advertisement within k hops. Any non-cluster-head sensor that receives such advertisements joins the cluster of the closest cluster head. Any sensor not associated with a cluster within t units of time becomes a forced cluster head.

Based on results in [25], the average energy consumed by the system in each time unit, $E[C]$, is derived as follows: $E[C] = 4\lambda a^2 \left(\frac{1-p}{2r\sqrt{p\lambda}} + \frac{0.756pa}{r} \right)$, where a is half of the area width, λ is the intensity of the spatial Poisson process with which sensors are distributed, and r is the transmission range. With the above equation, the optimal p can be obtained explicitly. *Bandyopadhyay et al. [2], [3] further determined the optimum value of p to minimize energy consumption in an h-level hierarchical sensor network*.

In summary, the schemes in [2] and [3] focus mainly on reducing the communication cost between sensors and their cluster heads. They have the following major characteristics: (1) the cluster heads may run out of energy before the other sensors, but the authors did briefly mention two mechanisms to balance the energy consumption; (2) no clock synchronization is needed; (3) sensors do not need to know location or distance information.

D. Energy-Efficient Surveillance System (ESS)

He et al. [15] designed the Energy-Efficient Surveillance System (ESS), in which the trade off between energy consumption and surveillance performance is explored by adaptively adjusting the sensitivity of the system. In [15], sensors are classified into sentries and non-sentries in each cycle. The sentries (similar to cluster heads) are elected locally by each sensor, using the information gathered from its neighbors. A sensor decides to become a sentry if it is an internal node of a diffusion tree or if it discovers that none of its neighbors is a sentry (or is covered by a sentry). A random backoff delay is used to avoid collisions when multiple sensors in the same neighborhood advertise their intent at the same time. The backoff delay of a sensor is set inversely proportional to its residual energy for balanced energy consumption [15].

The non-sentry sensors alternate between sleep and wake-up states. Two different schemes, proactive control and reactive control, to determine the sleep-wakeup cycle are described in [15]. In proactive control, the sentry sends out sleep beacons periodically. A non-sentry sensor stays awake until it receives a beacon from its sentry, and then sleeps for the sleep duration specified in the beacon. In the reactive control, the sentries do not send out explicit beacons, and each non-sentry sensor remains awake for awake-Duration and then sleeps for sleep-Duration. A non-sentry sensor can also be awaken by an awake beacon from a sentry. The reactive scheme is apparently more stealthy compared to the proactive scheme. Its drawback is that the clocks of the non-sentry sensors may drift in course of time, and as a result, a sentry may need to transmit an awake beacon repeatedly to wake up the non-sentries.

In summary, ESS saves energy by putting the non-sentry nodes to sleep most of the time. It has the following major characteristics: (1) the sentries are rotated (with randomization) to achieve balanced energy consumption; (2) clock synchronization is required at the beginning of each cycle; (3) sensors do not need to know location or distance information. (4) it assumes that the transceiver's power level can be adjusted to achieve certain transmission range.

E. Linear Distance-based Scheduling (LDS)

Deng et al. [12] proposed a sleep-scheduling algorithm, called Linear Distance-based Scheduling (LDS) scheme for cluster-based high density sensor networks. The goal is to reduce energy consumption while maintaining adequate sensing coverage capabilities [12]. *To achieve this goal, the LDS scheme selects sensors farther away from the cluster head to sleep with higher probabilities*. The rationale behind this scheme is based on the assumption that each sensor's radio transceiver is capable of changing its transmission power in continuous steps to achieve different transmission ranges; a farther away sensor needs more power to communicate with the cluster head, and therefore, has higher energy consumption. The LDS scheme only considers static clusters. In other words, cluster heads are not changed once they are selected.

In summary, LDS selects sensors to sleep according to their relative distances to the cluster head. It has the following major characteristics: (1) a sensor does not need to know other sensors' location information; (2) the scheme may cause uneven sensing coverage; in other words, a location farther away from the cluster head has less sensing coverage than a location closer to the cluster header; (3) the scheme may cause uneven lifetime in the cluster, i.e., sensors farther away from the cluster header live longer than sensors closer to the cluster header.

F. Balanced-energy Sleep Scheduling (BS)

Deng et al. proposed Balanced-energy Sleep Scheduling (BS) in [11]. BS extends the LDS scheme by evenly distributing the sensing and communication tasks among the non-head sensors so that their energy consumption is similar regardless of their distance to the cluster-head. More specifically, the authors derived a sleep probability function $p(x)$ so that the total energy consumption of a sensor does not depend on x, the distance between the sensor and its cluster head. The other aspects of BS are the same as LDS, so we do not summarize them here.

VI. Classification

Since there are many dimensions along which to classify the mechanisms, we do not attempt to enumerate all the possible combinations. Rather, we summarize the choices made by the surveyed mechanism and this summary can be used as an input to any classification system. We found that it is not easy to extracting the design assumptions and objectives from the papers. The designers usually state explicitly the most important assumptions and objectives, but we have to extrapolate the remaining ones and our extrapolation may not reflect the designers' original intentions accurately. To minimize this problem, we do not include in our summary the high-level design objectives, i.e., robustness, simplicity and scalability, which are more difficult to quantify.

Our results are presented in Table I and Table II. In Table I, high sensor density means that the total number of sensors is several orders of magnitude higher than the number of working nodes. In table II, we consider a mechanism to achieve stealthiness if it satisfies one of the following conditions: (a) no communication is required for sleep scheduling. For example, in the RIS mechanism, each node decides whether to go to sleep using a pre-determined probability; and (b) the operation is divided into rounds and communication among sensors to set up schedules is required only at the beginning of a round. Moreover, a mechanism is considered to achieve balanced energy consumption if sensors rotate to become on-duty and off-duty.

VII. Conclusions

We have reviewed 15 energy-efficient scheduling mechanisms that are designed specifically for sensor networks. We examined their design assumptions and objectives to find their commonalities and differences. We found that designers have to make many design assumptions about sensors' capabilities, sensor network structure and sensor deployment strategy, either explicitly or implicitly. Their design objectives also vary from simply maintaining sensing coverage to satisfying several QoS objectives simultaneously. We believe that it is important to design a sensor network that can support a wide range of applications and such a network would have only one or a few generic scheduling mechanisms that can be tailored to different applications. In our future work, we plan to study the feasibility of this approach.

Appendix

In this appendix, we summarize major theoretical results on sensor coverage and connectivity, which can be used in both initial sensor deployment and the dynamic scheduling of sleeping sensors.

TABLE I
CLASSIFICATION BASED ON ASSUMPTIONS

Schemes	Assumptions											
	Network Structure	Sensor Placement	High Sensor Density	Determi-nistic Detection	Sensing Area	Uniform Sensing Range	Adjustable Transmission Range	Time Synch.	Frequent Failures	Mobility	Known Location	Known Distance
RIS (Kumar)	Flat	Grid, Uniform, Poisson	No	Yes	2-D Disk	Yes	No	Yes	No	No	No	No
Sponsored Sector	Flat	Any	No	Yes	2-D Disk	Yes	No	Yes	No	No	Yes	No
MSNL	Flat	Any	No	Yes	2-D Convex	Any	No	No	No	No	Yes	No
LDAS	Flat	Uniform	No	Yes	2-D Disk	Yes	No	No	No	No	No	No
PEAS	Flat	Uniform	Yes	Yes	Any	Any	Yes	No	Yes	No	No	No
OGDC	Flat	Any	No	Yes	2-D Disk	Yes	No	Yes	No	No	Yes	No
CCP	Flat	Any	No	Yes	2-D Disk	Yes	No	No	No	No	Yes	No
ASCENT	Flat	Any	Yes	Yes	Any	Any	No	No	No	No	No	No
PECAS	Flat	Uniform	No	Yes	Any	Any	Yes	No	Yes	No	No	No
LEACH	Hierarchical	Any	No	Yes	Any	Any	No	Yes	No	No	No	No
E-LEACH	Hierarchical	Any	No	Yes	Any	Any	No	Yes	No	No	No	No
Bandyo-padhyay et al. [2], [3]	Hierarchical	Poisson	No	Yes	2-D Disk	Yes	No	Yes	No	No	No	No
ESS	Hierarchical	Any	No	Yes	Any	Any	Yes	Yes	No	No	No	No
LDS	Hierarchical	Poisson	No	Yes	2-D Disk	Yes	Yes	Yes	No	No	No	Yes
BS	Hierarchical	Poisson	No	Yes	2-D Disk	Yes	Yes	Yes	No	No	No	Yes

TABLE II
CLASSIFICATION BASED ON OBJECTIVES

Schemes	Objectives				
	Sensing Coverage Guarantee	Network Connectivity Guarantee	Data Delivery Ratio	Stealthiness	Balanced Energy Consumption
RIS (Kumar)	Full, K, Asymptotic			Yes	Yes
Sponsored Sector	Full, Original, Hard			Yes	Yes
MSNL	Full, K, Hard				Yes
LDAS	Partial, 1, Statistical				Yes
PEAS		1, Hard			
OGDC	Full, 1, Hard	1, Hard		Yes	Yes
CCP	Full, K, Hard	K, Hard			Yes
ASCENT			Yes		
PECAS		1, Hard			Yes
LEACH				Yes	Yes
E-LEACH				Yes	Yes
Bandyo-padhyay et al. [2], [3]				Yes	
ESS				Yes	Yes
LDS					
BS					Yes

A. *K-Coverage in 2-D Space*

Huang et al. proved a sufficient and necessary condition for k-coverage in a two-dimensional area [19], [20]. A sensor is k-perimeter-covered if all points on the perimeter of the sensor are covered by at least k sensors other than itself [20]. Huang et al. derived the following relationship between k-perimeter-coverage and k-coverage [20]:

Theorem 1: Suppose that no two sensors are located in the same location. The whole network area A is k-covered iff each sensor in the network is k-perimeter-covered.

Note that the above theorem is true only if *all* sensors are k-perimeter-covered. This theorem holds whether the sensing range is a unit disk or a non-unit disk.

B. *K-Coverage in 3-D Space*

Huang et al. also extended the above theorem to three-dimensional space [21]. They defined the k-coverage problem in 3-D space as follows: given n sensors $S = \{s_1, s_2, ..., s_n\}$, a sensing range of r_i and each sensor s_i's location (x_i, y_i, z_i) in a three-dimensional cuboid sensing field A, the problem is to determine whether all points in A are k-covered under unit disk assumption. Huang el. al. proved the following theorem in [21]:

Theorem 2: A sphere is k-sphere-covered if each subspace that is adjacent to the sphere is covered by at least k sensors other than itself. If each sphere is k-sphere-covered, then the sensing field A is k-covered.

The above two theorems provide a foundation for evaluating whether a sensing field (or space) is k-covered after initial sensor deployment. Furthermore, they can be used by a distributed scheduling mechanism to turn off sensors – each sensor just needs to make sure that *all* its neighbors are still k-perimeter-covered (or k-sphere-covered) when it is turned off. However, such a mechanism requires that each sensor knows its own location and its neighbors' locations.

C. Asymptotic K-Coverage

Assuming that active sensors are self-elected with a probability p using the Random Independent Scheduling (RIS) mechanism, Kumar et al. derived the sufficient conditions for achieving *asymptotic k-coverage* [22]. Three deployment strategies are studied: grid ($\sqrt{n} \times \sqrt{n}$), random uniform (n points), and 2-dimensional Poisson (with rate n). Due to space constraint, we only cite their results for the random uniform deployment here as the other results are similar.

Theorem 3: Let n sensors be deployed uniformly over a unit square region. If, for some slowly growing $\phi(np)$, the probability p and the sensing range r satisfy

$$c(n) = 1 + \frac{\phi(np) + k \log\log(np)}{\log(np)} \tag{1}$$

for sufficiently large n, then the unit square region is almost always k-covered.

D. Complete and Partial Redundancy in Coverage

In [30], Wu et al. studied the probabilistic conditions for *complete redundancy*, i.e., when a sensor's sensing area is completely covered by its neighbors' sensing areas. Then they studied the conditions for partial redundancy. They proved the following theorems under the assumption that *sensors are randomly and uniformly deployed*.

Theorem 4: Let C be the sensing area of a sensor. Given C and its neighboring sensing areas C_i's ($1 \leq i \leq n$), if A is the event that C is fully covered by C_i's, then $1 - n0.609^{n-1} \leq Pr\{A\} \leq 1 - n0.609^{n-1} + \epsilon$, where $\epsilon = \frac{n(n-1)}{2}(0.276)^{n-1}$.

Theorem 5: The average percentage of a sensor's sensing area that is covered by its n random neighbors is not smaller than $1 - 0.609^n - (\frac{n}{6} - 0.109)0.109^{n-1}$.

They then concluded that it is much more expensive to turn off nodes based on complete redundancy than partial redundancy [1]. Based on these results, they proposed the Lightweight Deployment-Aware Scheduling Mechanism (LDAS) to maintain statistical partial coverage [30].

E. 1-*Connectivity*

Zhang and Hou [34] proved that if a sensor network provides complete sensing coverage, it also provides full connectivity when it satisfies the condition that *the radio transmission range is at least twice the sensing range*. This result implies that, if the radio transmission range of sensors can be configured to satisfy this condition, a sleep-scheduling mechanism only needs to maintain sensing coverage in order to provide both coverage and connectivity. Based on this result, Zhang and Hou then proposed the Optimal Geographic Density Control (OGDC) mechanism to maintain both 1-coverage and 1-connectivity [34]. Wang et al. proved a more general form of this theorem in [29] (also see Section F).

[1]Wu et al. arrived at this conclusion based on this observation: at least 11 neighbors will be required to get 100% redundancy (complete redundancy) *with a probability of 90%*, whereas only 5 neighbors can provide at least 90% partial redundancy *on average* [30].

F. K-Connectivity

Wang et al. discovered the following relationship between K-coverage and K-connectivity in [29]:

Theorem 6: A set of nodes that K-cover a convex region A forms a K connected communication graph if $R_c \geq 2R_s$, where R_c is the radio transmission range and R_s is the sensing range. For a set of sensors that K-cover a convex region A, the interior connectivity is $2K$ if $R_c \geq 2R_s$.

An obvious implication of the above theorem is that, when the radio transmission range is at least twice the sensing range, i.e., $R_c \geq 2R_s$, any scheduling protocol can ensure that the boundary nodes in the network are K-connected and the interior nodes are $2K$-connected by maintaining a sensing coverage degree of K. This result is used by the Coverage Configuration Protocol (CCP) proposed by Wang et al. [29].

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

[2] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM*, 2003.

[3] S. Bandyopadhyay and E. J. Coyle. Minimizing communication costs in hierarchically-clustered networks of wireless sensors. *Computer Networks*, 44:1–16, 2004.

[4] P. Berman, G. Calinescu, C.Shah, and A. Zelikovsly. Efficient energy management in sensor networks. In Y. Xiao and Y. Pan, editors, *Ad Hoc and Sensor Networks*. Nova Science Publishers, 2005.

[5] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[6] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM 2002*, New York, NY, June 2002.

[7] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[8] Crossbow. Mpr/mib mote hardware users manual. `http://www.xbow.com/Support/manuals.htm`, aug 2004.

[9] Crossbow. Power management and batteries. `http://www.xbow.com/Support/appnotes.htm`, 2004.

[10] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney. Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks. *Elsevier Computer Communications Journal, Special Issue on ASWN '04*, 2004. to appear.

[11] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney. Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks. In *Proc. of the 4th Workshop on Applications and Services in Wireless Networks (ASWN '04)*, 2004.

[12] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney. Scheduling sleeping nodes in high density cluster-based sensor networks. *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on "Energy Constraints and Lifetime Performance in Wireless Sensor Networks"*, 2004. to appear.

[13] J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS '01)*, 2001.

[14] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom '04)*, pages 129–143, 2004.

[15] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*, pages 270–283, 2004.

[16] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-speciffic protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, Oct. 2002.

[17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of Hawaaian International Conference on Systems Science*, Jan. 2000.
[18] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proceedings of Ninth International Conference on Architectural Support for Porgramming Languages and Operating Systems*, Nov. 2000.
[19] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, pages 115–121, 2003.
[20] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. *ACM Mobile Networks and Applications (MONET), special issue on Wireless Sensor Networks*, 2005. to appear.
[21] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo. The coverage problem in three-dimensional wireless sensor networks. In *Proceedings of IEEE GLOBECOM*, 2004.
[22] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom '04)*, pages 144–158, 2004.
[23] M. J. Miller and N. H. Vaidya. A mac protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on Mobile Computing*, 4(3):228–242, May/June 2005.
[24] R. Powers. Batteries for low power electronics. *Proceedings of the IEEE*, pages 687–693, April 1995.
[25] S.G.Foss and S. Zuyev. On a voronoi aggregative process related to a bivariate poisson process. *Advances in Applied Probability*, 28(4):965–981, 1996.
[26] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of ACM Mobicom '01*, pages 272–287, 2001.
[27] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *IEE ICC 2001*, Helsinki, Finland, June 2001.
[28] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 32–41, 2002.
[29] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys '03)*, pages 28–39, 2003.
[30] K. Wu, Y. Gao, F. Li, and Y. Xiao. Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on "Energy Constraints and Lifetime Performance in Wireless Sensor Networks"*, 10(6):837 – 852, Dec. 2005.
[31] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan. CoGrid: an efficient coverage maintenance protocol for distributed sensor networks. In *Proceedings of IPSN 2004)*, Berkeley, CA, Apr. 2004.
[32] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS '03)*, pages 28–37, 2003.
[33] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM 2002*, pages 1567–1576, New York, NY, June 2002.
[34] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. In *Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004. invited paper.

Lan Wang is an assistant professor in the Department of Computer Science at the University of Memphis. She holds a B.S. degree (1997) in Computer Science from Peking University, China and a Ph.D. degree (2004) in Computer Science from University of California, Los Angeles. Lan's research interests include Internet architecture and design, network security, routing, sensor networks, network performance measurement and large-scale simulation. She has served on the TPC of several conferences. She's a member of the IEEE and the ACM. For more information, please visit `http://www.cs.memphis.edu/~lanwang/`.

Yang Xiao worked at Micro Linear as a MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined the Department of Computer Science at The University of Memphis in 2002. Dr. Xiao was a voting member of the IEEE 802.11 Working Group from 2001 to 2004. He currently serves as editor-in-chief for *International Journal of Security and Networks (IJSN)* and for *International Journal of Sensor Networks (IJSNet)*. He serves as an associate editor or on editorial boards for the following refereed journals: *(Wiley) International Journal of Communication Systems*, *(Wiley) Wireless Communications and Mobile Computing*, *EURASIP Journal on Wireless Communications and Networking*, and *International Journal of Wireless and Mobile Computing*. He has served as a lead/sole journal guest editor for five journals during 2004-2005. He serves as TPC member for more than 60 conferences including ICDCS, ICC, GLOBECOM, WCNC, etc. He serves as a referee for many funding agencies, as well as a panelist for the US NSF. His research areas include wireless networks, mobile computing, and network security. Email: yangxiao@ieee.org.